#3

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re U.S. Patent Application of	)
	)
TARUI et al.	)
	)
Application Number: To Be Assigned	)
	)
Filed: Concurrently Herewith	)
	)
For: A COMPUTER FORMING LOGICAL PARTITIONS	)
	)

Honorable Assistant Commissioner
for Patents
Washington, D.C. 20231

REQUEST FOR PRIORITY
UNDER 35 U.S.C. § 119
AND THE INTERNATIONAL CONVENTION

Sir:

In the matter of the above-captioned application for a United States patent, notice is hereby given that the Applicant claims the priority date of January 24, 2001, the filing date of the corresponding Japanese patent application 2001-015196.

The certified copy of corresponding Japanese patent application 2001-015196 is being submitted herewith. Acknowledgment of receipt of the certified copies is respectfully requested in due course.

Respectfully submitted,

Stanley P. Fisher
Registration Number 24,344

JUAN CARLOS A. MARQUEZ
Registration No. 34,072

REED SMITH HAZEL & THOMAS LLP
3110 Fairview Park Drive
Suite 1400
Falls Church, Virginia 22042
(703) 641-4200
August 30, 2001

日　本　国　特　許　庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日
Date of Application: ２００１年　１月２４日

出願番号
Application Number: 特願２００１－０１５１９６

出願人
Applicant(s): 株式会社日立製作所

２００１年　５月１１日

特許庁長官
Commissioner,
Japan Patent Office 及川耕造

出証番号　出証特２００１－３０３７７４６

【書類名】　特許願

【整理番号】　H00013141A

【あて先】　特許庁長官　殿

【国際特許分類】　G06F 15/177

【発明者】

【住所又は居所】　東京都国分寺市東恋ケ窪一丁目２８０番地　株式会社日立製作所中央研究所内

【氏名】　垂井　俊明

【発明者】

【住所又は居所】　東京都国分寺市東恋ケ窪一丁目２８０番地　株式会社日立製作所中央研究所内

【氏名】　亀山　伸

【発明者】

【住所又は居所】　東京都国分寺市東恋ケ窪一丁目２８０番地　株式会社日立製作所中央研究所内

【氏名】　マシエル　フレデリコ

【発明者】

【住所又は居所】　東京都国分寺市東恋ケ窪一丁目２８０番地　株式会社日立製作所中央研究所内

【氏名】　庄内　亨

【特許出願人】

【識別番号】　000005108

【氏名又は名称】　株式会社　日立製作所

【代理人】

【識別番号】　100075096

【弁理士】

【氏名又は名称】　作田　康夫

【電話番号】　03-3212-1111

【手数料の表示】

【予納台帳番号】　013088

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【プルーフの要否】　要

【書類名】明細書

【発明の名称】計算機およびその入出力手段

【特許請求の範囲】

【請求項１】

１つ以上のＣＰＵ、主記憶、１つ以上の入出力手段からなり、複数のパーティションに分割可能な計算機において、入出力手段のパーティションへの割当てを制御する手段を設けたことを特徴とする計算機。

【請求項２】

各パーティションに対応して、上記パーティションの入出力性能を測定する手段を設けたことを特徴とする請求項１記載の計算機。

【請求項３】

オペレータが、各パーティションの入出力割当てを指示する手段を設けたことを特徴とする請求項１記載の計算機。

【請求項４】

各パーティションへの入出力割当てを予約する手段を設けたことを特徴とする請求項１記載の計算機。

【請求項５】

各パーティションの入出力性能と、あらかじめ決められた入出力割合の変更条件を比較する手段、上記入出力割当て変更条件が満たされた場合、上記パーティションの入出力割当てをオペレータの介在無しに変更する手段を設けたことを特徴とする請求項２記載の計算機。

【請求項６】

パーティションの入出力割当てを増加させた時間を記録する手段、上記記録に基づき、上記パーティションのユーザへの課金を割り増す手段を設けたことを特徴とする請求項５記載の計算機。

【請求項７】

各パーティションの処理性能と、ＳＬＡ（サービス・レベル・アグリーメント）に基づき、あらかじめ決められたパーティションの下限性能とを比較する手段、上記性能が下限性能を下回った場合、もしくは下回る可能性がある場合、パー

ティションのＣＰＵ性能、入出力性能より、ＣＰＵネックであるか入出力ネックであるかを判定する手段、入出力ネックと判定され、かつ他のパーティションの入出力性能に余裕がある場合、上記パーティションへの入出力割当量を増加させる手段を設けたことを特徴とする請求項２記載の計算機。

【請求項８】

入出力ネックであり、他のパーティションの入出力性能に余裕がない場合、ＳＬＡが守られなかったことを記録する手段、上記記録に基づき上記パーティションユーザへの課金を割り引く手段を設けたことを特徴とする請求項７記載の計算機。

【請求項９】

入出力性能のモニタ結果を、上記計算機外部の第２の計算機に伝える手段、上記第２の計算機で行われたＳＬＡの判定および入出力割当ての変更要求に基づき、入出力割当てを変更する手段を設けたことを特徴とする請求項７の計算機。

【請求項１０】

各パーティションの入出力割当量を、上記パーティションへのＣＰＵ割当量と比例させて変更する手段を設けたことを特徴とする請求項１記載の計算機。

【請求項１１】

各パーティションの性能を測定する手段、上記測定結果とユーザがあらかじめ設定した条件に基づいて、パーティションへの入出力割当てを変更することを特徴とする請求項１記載の計算機。

【請求項１２】

第１のパーティションが行っている通信を、あらかじめ決められた大きさのデータを通信した後に中断する手段、上記中断後に他のパーティションが要求する通信に切り替える手段、上記他のパーティションの通信があらかじめ決められた大きさのデータを送った後で、第１のパーティションの通信を再開する手段を設けたことを特徴とする請求項１記載の計算機。

【請求項１３】

各パーティションがアクセス可能な入出力アダプタを動的に変更する手段を設けたことを特徴とする請求項１の計算機。

【請求項１４】

１つ以上のＣＰＵ、主記憶、１つ以上の入出力手段からなり、複数のパーティションに分割可能な計算機のための入出力手段であって、外部から指定された割合で各パーティションの入出力処理を実行することを特徴とする計算機の入出力手段。

【請求項１５】

１つ以上のＣＰＵ、主記憶、１つ以上の入出力手段からなり、複数のパーティションに分割可能な計算機のための入出力手段であって、上記入出力手段をアクセス可能なパーティション番号と、上記パーティションが上記入出力手段をアクセスするために使用するベースアドレスの組を複数記憶する設定レジスタを持ち、上記設定レジスタをパーティション制御プログラムが動的に設定する手段を持つことを特徴とする計算機の入出力手段。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は情報処理装置、特に、サーバ、メインフレーム等に用いられる、１台の計算機内部に複数のパーティションを動作させることができる計算機システムおよびその入出力手段に関する。

【０００２】

【従来の技術】

複数のシステムのコンソリデーションによるＴＣＯの削減、ホットスタンバイによる可用性の向上等を目的として、１台の計算機システムを複数のパーティションに分け、異なるパーティションに別々のＯＳを乗せ、あたかも物理的に複数の計算機システムが有るかのように見せるシステムが広く用いられている。

【０００３】

特開平６－３５７２５においては、汎用計算機におけるロジカルパーティション（ＬＰＡＲ）技術が開示されている。同技術によれば、ハイパバイザと呼ばれるパーティション制御プログラムにより、複数のパーティションによりＣＰＵを時分割で使用することにより、ＣＰＵ台数を超えるパーティションを１つの計算

機システム上に動作させ、コンソリデーションを実現することが可能である。さらに、各パーティションに割当てるＣＰＵ時間をハイパバイザにより制御することを可能にする。さらに、Ｉ／Ｏのパーティション間共有を実現する。

【０００４】

特開２０００－１３２５３０においては、主記憶共有型の並列計算機システムを複数のＣＰＵのグループに分け、各々のグループを異なるパーティションとして使用する技術が開示されている。同技術によれば、並列計算機の内部を複数のパーティションに分け、各パーティションに物理的なＣＰＵ、主記憶、Ｉ／Ｏを割当てることを可能にする。

【０００５】

近年、計算機の入出力方式の分野では計算機からの通信、Ｉ／Ｏの新しい方式が提案されている。コンパック（Ｃｏｍｐａｑ）、インテル（Ｉｎｔｅｌ）、マイクロソフト（Ｍｉｃｒｏｓｏｆｔ）の３社は「バーチャル・インタフェース・アーキテクチャ・スペシフィケーション（Virtual Interface Architecture Specification）」において、ＶＩＡプロトコルと言う新しい通信プロトコルを提案している。同プロトコルにおいては、下記の２つの技術により、従来の通信方式と比べて大幅に通信オーバヘッドを削減し、高速な通信、Ｉ／Ｏを実現する。

（１）通信起動時にＯＳへのシステムコールを行わずに、ユーザプログラムから直接通信を起動することを可能にする（ＯＳレス起動）。

（２）通信データをユーザ空間の通信バッファから、ＯＳ空間の通信バッファにコピーすることなく、ユーザ空間から直接通信アダプタがコピーすることを可能にする（コピーレス起動）。

【０００６】

さらに、ＶＩＡによる通信を核プロトコルとする、次世代のＩ／Ｏアーキテクチャが提案されている。従って、将来の計算機間の通信、Ｉ／Ｏアーキテクチャを考える場合、ＶＩＡのようなＯＳレス、コピーレス起動が主流になると考えられる。

【０００７】

【発明が解決しようとする課題】

しかしながら、パーティションに分けられた計算機において、上記従来技術を用いてＩ／Ｏを行おうとした場合、下記の問題がある。

【０００８】

各パーティションのＣＰＵリソースの使用量に関しては柔軟に制御する手段が用意されているが、Ｉ／Ｏリソースの使用量は制御することは困難である。

【０００９】

大型計算機のパーティションにおいては、各パーティションがＣＰＵの何％使用できるかを、パーティション制御プログラムにより制御することが可能である。また、並列計算機のパーティションにおいても、各パーティションにＣＰＵを何個割当てるかを制御することができる。しかし、各パーティションの入出力に関しては、パーティションにある特定の入出力アダプタを割当てることが普通であり、Ｉ／Ｏアダプタの共有を許しても、各パーティションへの割当てはベストエフォート（best　effort）であり、Ｉ／Ｏの割当てを積極的に制御することは行われてこなかった。

【００１０】

旧来のＩ／Ｏ方式では、Ｉ／Ｏを行うにはＩ／Ｏを起動するためのシステムコール、データ領域のコピー等、かなりのＣＰＵパワーを必要とするため、ＣＰＵパワーさえ制御しておけば、Ｉ／ＯにもＣＰＵとほぼ比例した能力を割当てることができた。

【００１１】

ところが、近年実用化されつつある、次世代のＩ／Ｏ方式においては、Ｉ／Ｏの起動はＯＳレス、コピーレスで行われるため、ＣＰＵパワーをほとんど使用せずにＩ／Ｏを起動することが可能になる。従って、ＣＰＵパワーを制御することにより、間接的にＩ／Ｏの使用量を制御すると言う従来の方式はうまく働かなくなりつつある。

【００１２】

さらに、近年飛躍的に重要さを増しつつあるインタネット関連のプログラムにおいては、従来のＣＰＵ主体のプログラムと比較して、格段に大きいＩ／Ｏ能力

を要求するプログラムが存在する。そのようなプログラムにおいては、従来考えられていたように、ＣＰＵ能力と比例したＩ／Ｏ能力を割当てておけばよいと言う考えでは、ＣＰＵ能力とＩ／Ｏ能力のバランスをとることが困難になりつつある。

【００１３】

さらに、現在のインタネットデータセンタのサーバにおいては、ユーザに対するＳＬＡ（サービス・レベル・アグリーメント）を守ることが必須の機能である。パーティション上で動作するユーザプログラムのＳＬＡを守るためには、パーティションのＣＰＵ使用量の他に、Ｉ／Ｏ使用量を制御する機能は必須である。

【００１４】

従って、本発明の目的は、パーティションを持った計算機システムにおいて、パーティションが使用するＩ／Ｏ能力をＣＰＵ割当量とは独立に制御する計算機およびその入出力手段を提供することにある。また、本発明の他の目的は、各パーティションに割当てられているＩ／Ｏ能力を制御することにより、各パーティション上のプログラムのＳＬＡを保持することにある。

【００１５】

また、近年のネットワーク、Ｉ／Ｏの高速化に伴い、１つのＩ／Ｏアダプタを１つのパーティションに占有させると、Ｉ／Ｏの性能を十分活かすことがでず、効率が悪い。

【００１６】

本発明のさらに他の目的は、複数のパーティションが１つのＩ／Ｏアダプタを共有した場合、複数のＩ／Ｏアダプタを各パーティションに割当てた場合、双方において各パーティションへのＩ／Ｏ能力の割当てを制御することである。

【００１７】

【課題を解決するための手段】

本発明においては上記目的を達成するために、パーティションに分けられた計算機において、

・オペレータが各パーティションへのＩ／Ｏ割当てを指示する手段、

・各パーティションへのＩ／Ｏ割当てをあらかじめ設定ファイルにより予約する

手段、

・各パーティションのＩ／Ｏ性能を測定する手段と、Ｉ／Ｏ性能がある決められた値を下回った場合、パーティション制御プログラムが自動的にＩ／Ｏ割当てを変更する手段、

・Ｉ／Ｏアダプタの各パーティションへの割当て量を制御するために、各パーティションのユーザプログラムの性能がＳＬＡで決められた値を下回った場合で、ユーザプログラムの性能低下がＩ／Ｏネックにより生じたと判断された場合、パーティション制御プログラムが自動的にＩ／Ｏ割当てを変更する手段

を具備する。

【００１８】

さらに、本発明による他の望ましい態様では、各Ｉ／Ｏアダプタにおいて、Ｉ／Ｏアダプタがある第１のパーティションのＩ／Ｏをあらかじめ決められた量だけ行った後に、Ｉ／Ｏを一時中断する手段、その後、他のパーティションのＩ／Ｏをそれぞれある決められた量だけ実行させる手段、他のパーティションのＩ／Ｏが終わった後に、第１のパーティションのＩ／Ｏを中断した部分より再開する手段を設ける。それにより、各パーティションが上記Ｉ／Ｏアダプタを使用できる時間を時分割で制御することを可能にする。時分割のパラメータ（各Ｉ／Ｏアダプタが連続してＩ／Ｏできる量）を制御することにより、パーティションへのＩ／Ｏ割当て量を変更することを可能にする。

【００１９】

さらに、本発明による他の望ましい態様では、複数のＩ／Ｏアダプタを持つ計算機において、各パーティションにどのＩ／Ｏアダプタを割当てることができるかを動的に制御する手段を設け、パーティションに割当てるＩ／Ｏ能力を制御することを可能にする。

【００２０】

【発明の実施の形態】

図１は本発明に係る計算機のブロック図である。計算機はＣＰＵ１０～１２、チップセット２０、主記憶３０、Ｉ／Ｏアダプタ１００～１０１からなっている。Ｉ／Ｏアダプタ１００～１０１は、外部のＩ／Ｏネットワーク１００ａに接続

されている。以下ではＩ／Ｏアダプタ１００の中のみ詳細に記す。他のＩ／Ｏアダプタの内部も同じ構成である。

【００２１】

Ｉ／Ｏアダプタ１００の内部で、１１０はどのパーティションが上記Ｉ／Ｏアダプタをアクセスしたかを判断するアクセスパーティション判定回路である。Ｉ／Ｏアダプタがどのパーティションのアクセスを扱うかは、本回路１１０により制御される。１２０はＣＰＵからのアクセスを受け付けるためのコントロールレジスタ群、１３０、１８０はＣＰＵアクセスに呼応して実際の入出力処理（データのネットワークへの出し入れ）を行う送信・受信回路である。２００は、送信・受信回路１３０、１８０が主記憶３０上のデータを論理アドレスでアクセスすることを可能にするためのアドレス変換回路である。

【００２２】

ユーザプログラムは、Ｉ／Ｏアダプタ１００～１０１に対して、論理アドレスで入出力を指示するため、Ｉ／Ｏアダプタ内部で論理／物理アドレス変換を行わなければならない。ユーザプログラムによるコントロールレジスタ群１２０に対する入出力の指示方法、送信回路１３０、受信回路１８０における入出力処理そのものについては、従来の入出力アダプタと同様であるので詳細は略す。

【００２３】

１４０、１９０は送信・受信スケジューラであり、上記Ｉ／Ｏアダプタが複数のパーティションの入出力を実行するときに、各パーティションのＩ／Ｏをどのような割合で実行するかを制御する。１５０は送信・受信スケジューラ１４０、１９０に各パーティションの入出力割当てを指示する送受信割当てレジスタである。１６０は受信したＩ／Ｏメッセージのパーティションを判定する回路、１７０は受信メッセージバッファである。さらに、各パーティションのＩ／Ｏ性能をハードウェアで測定するためのモニタリング手段２１０を持つ。

【００２４】

図２に本発明が実現するＩ／Ｏ割当て方式を示す。図２はシステム内部に２つのパーティションが存在する場合である。パーティションのＣＰＵ処理能力は同図（ａ）の３００、３０１に示すように、各々のパーティションに５０％づつ割

り当ててある。Ｉ／Ｏ割当て方法としては大きく分けて同図（ｂ）のような時分割割当ておよび同図（ｃ）のような空間分割割当ての２つの方法がある。

【００２５】

上記一方の時分割割当てにおいては、１つのＩ／Ｏアダプタを２つのパーティションが時分割で共用している。当初は２つのパーティションのＩ／Ｏ処理能力３１０、３１１は均等に分割されている。ここで、パーティション０の入出力能力を増加する必要が生じた場合、時分割の割当てを変更することにより、パーティションのＩ／Ｏ処理能力の割当てを５０％づつ（３１５）から、パーティション０は７５％、パーティション１は２５％に変更する（３１６）。

【００２６】

他方、空間分割割当ては、システム内の複数のＩ／Ｏアダプタ３２０～３２３を各パーティションがどのように使うかを制御することにより達成される。各々のＩ／Ｏアダプタは外部にある同一のＩ／Ｏスイッチに接続され、全てのアダプタが同一の機器に通信を行なうことが可能であることが前提である。当初は３３１、３３２に示すようにパーティション０、パーティション１は２個づつのＩ／Ｏアダプタを使用しており、各パーティションのＩ／Ｏ処理能力は５０％で同一である。ここで、パーティション０の入出力能力を増加する必要が生じた場合、今までパーティション１が使用していたアダプタ２（３２２）をパーティション０が使用するように割当て直すことにより（境界は３３２、３３３となる）、各パーティションへのＩ／Ｏ能力の割当てを、パーティション０は７５％、パーティション１は２５％に変更する。

【００２７】

上記の時分割および空間分割は、それぞれ単独で使うほかに、時分割と空間分割を組み合わせて使うように制御することも可能である。例えば、パーティション０はアダプタ２．５個、パーティション１はアダプタ１．５個を使用するように制御することもできる。

【００２８】

本発明の特徴は、Ｉ／Ｏアダプタ内の送受信割当てレジスタ１５０、送信スケジューラ１４０、受信スケジューラ１９０に、時分割割当てを制御する機能を持

たせ、アクセスパーティション判定回路１１０に空間分割を制御する機能を持たせ、パーティションとＩ／Ｏアダプタの関連を柔軟に制御することにより、各パーティションのＩ／Ｏ能力を図２で示すように柔軟に制御することである。

【００２９】

以下では図１、図２～図１６を用いて、本発明のＩ／Ｏアダプタの動作を詳細に説明する。以下の実施例では、説明を容易にするために、システムの最大パーティション数は４個であるが、最大パーティション数はハードウェア量の許す限り任意の個数可能である。さらに、レジスタ類を主記憶上にスワップする機能を設けることにより、アダプタの中にハードウェアで用意されているレジスタの数にとらわれずに、事実上無限個のパーティションを取り扱うことが可能である。

【００３０】

図１においてＣＰＵ１０～１２から出されたＩ／Ｏ要求は、まずアクセスパーティション判定回路１１０に入力され、アクセスされたアドレスより、どのパーティションからアクセスされたかを判断される。

【００３１】

図６にアクセスパーティション１１０判定回路の詳細（アドレス部分のみ）を示す。アクセスパーティション判定回路１１０は、パーティションのベースアドレスレジスタ１１００およびアドレス範囲判定回路１１５０により構成される。ベースアドレスレジスタ１１００は各パーティションに対応して、各パーティションが上記Ｉ／Ｏアダプタをアクセスする時に使用するベースアドレス（１１２０～１１２３）、および、各パーティションが上記Ｉ／Ｏアダプタをアクセスすることを許すかどうかを示すバリッド（ｖａｌｉｄ）ビット１１１０～１１２３を持ち、上記ｖａｌｉｄビットが１のパーティションがアクセスを許される。

【００３２】

ハイパバイザ、ＳＶＰ等のパーティション制御プログラムは、上記Ｉ／Ｏアダプタにアクセスを許すパーティションのｖａｌｉｄビットを１にし、ベースアドレスをセットする。複数のパーティションから共有されているアダプタに関しては、各々のパーティションのｖａｌｉｄビットを１にするとともに、各パーティションに対応して、異なるベースアドレスを指定する。各パーティションのＯＳ

は、ベースアドレスレジスタで示されたアドレスで上記Ｉ／Ｏアダプタをアクセスするようにコンフィグレーションされる。各パーティションに属するＩ／Ｏアダプタを変更する場合は、ベースアドレスレジスタ１１００を変更する。

【００３３】

ＣＰＵからのＩ／Ｏアクセス信号１１０ｂのうちのアドレス部分１１０ｂ１はアドレス範囲判定回路１１５０に入力される。アドレス判定回路１２００では、ｖａｌｉｄビットが１であるパーティションのエントリのうちで、ＣＰＵのアクセスアドレス１１０ｂ１が、ベースアドレスと、ベースアドレス＋上記Ｉ／Ｏアダプタの占有アドレス範囲の間に入るパーティションを判定する。該当するパーティションが見つかった場合には、パーティション番号１１０ａ２には上記で求めたパーティションの番号を出力するとともに、レジスタアドレス（１１０ａ１）においては、アクセスアドレスー上記パーティションのベースアドレスが出力される。上記の信号は、ＣＰＵアクセス信号の他の部分と共に、信号１１０ａを通じてコントロールレジスタ群１２０に伝えられる。

【００３４】

図７にコントロールレジスタ群１２０の内部構造を示す。コントロールレジスタ群１２０はアクセス分配回路１２００と各パーティションに対応するコントロールレジスタ１２１０～３により構成される。信号１１０ａを通じて伝えられたＣＰＵのアクセス信号は、パーティション番号１１０ａ２に応じて、該当するパーティションのコントロールレジスタ１２１０～３の何れかをアクセスする。パーティション毎にコントロールレジスタを持たせることにより、複数のパーティションからのＩ／Ｏ要求に対処することを可能にする。

【００３５】

コントロールレジスタ１２０に入力されたＩ／Ｏ要求は信号１２０ａを通じて送信回路１３０もしくは受信回路１８０に伝えられ、送受信処理が行われる。コントロールレジスタの詳細は従来技術と同等技術であるので、説明を略す。

【００３６】

送受信回路の動作の説明に先立って、本発明のシステムで使われるＩ／Ｏパケットの形式について説明する。ネットワーク上を流れるＩ／Ｏパケットの大きさ

は、ある決められたパケットサイズにより制限される。そのため、主記憶上のデータがパケットサイズより大きい場合には、データをパケットサイズに分割して送受信しなければならない。

【００３７】

図１３に主記憶上のＩ／Ｏデータ４１００とＩ／Ｏパケット４３００～４３０３の関係を示す。図の場合は４３００～４３０３の４つのパケットに分割されている。各パケットはヘッダ領域４３２０～４３２３、データ領域４３１０～４３１３を含む。送受信データのパケットへの分割、組み立て、パケットのヘッダ形式等については、従来技術と同等であるため、説明を略する。本発明の特徴としては、パケットのヘッダ領域の中に、従来からある項目に加えて、上記データを送受信するパーティションの番号４３３０～４３３３を表わすフィールドが追加されていることである。

【００３８】

図８に送信回路１３０の構造を示す。送信回路は、実際の送信処理（主記憶からのＤＭＡ）を行う送信側ＤＭＡＣ１３３０、および、各パーティションの送信の途中状態を示すレジスタ群、送信中のアドレスを示すレジスタ１３１０～１３１３、送信しなければならない残りのバイト数を示すバイトカウンタ１３２０～１３２３、により構成される。コントロールレジスタ群１２０からのＩ／Ｏ要求１２０ａは、送信が指示されたパーティションに該当する送信レジスタ群１３００に入力される。その後、送信側ＤＭＡＣ１３３０は送信スケジューラ１４０の指示により動作する。

【００３９】

図９に上記送信側ＤＭＡＣ１３３０の動作フローを示す。まず、送信スケジューラ１４０から信号１４０ａによって伝えられた送信要求１４０ａ３が入ると、送信スケジューラ１４０が指示する次に送信すべきパーティション番号１４０ａ４を読込む（ステップ５０００）。送信側ＤＭＡＣ１３３０は、送信状態レジスタ群１３００より、指示されたパーティションのアドレスレジスタ、バイトカウンタを読込む（ステップ５００１）。

【００４０】

バイトカウンタが０の場合は該当するパーティションに関しては送信するデータが無いため、送信終了報告ステップ５００４に飛ぶ（ステップ５００２）。バイトカウンタが０ではない場合、主記憶上の送信データを１パケット分送信した（ステップ５００３）後、送信スケジューラ１４０に信号１４０ａを通じて、１パケット分の送信が終了したことを報告（１４０ａ１）する（ステップ５００４）。

【００４１】

送信スケジューラ１４０は、パケットの切れ目で上記パーティションのデータの送信を中断して、他のパーティションのデータを送信する必要があると判断した場合、信号１４０ａを通じて送信中断を指示（１４０ａ２）する。中断指示１４０ａ２がない場合には、送信側ＤＭＡＣ１３３０はステップ５００３に戻り、該当するパーティションのデータを送信し続ける（ステップ５００５）。送信中断指示１４０ａ２がある場合には、データ送信の途中状態（アドレスレジスタ、バイトカウンタ）を、送信状態レジスタ群１３００の中の、該当するパーティションのエントリに書込む。その後、送信側ＤＭＡＣ１３３０はステップ５０００に戻り、次に送信すべきパーティションの番号を読込む。

【００４２】

ここで、ユーザプログラムからのデータ転送要求は論理アドレスを用いて行われるため、送信側ＤＭＡＣ１３３０が主記憶上の送信データをアクセスする場合には、論理アドレスが用いられる。従って、Ｉ／Ｏアダプタの中に、論理アドレスから物理アドレスに変換するアドレス変換手段２００が必要になる。送信回路１３０からは、信号２００ｂを通じて、アクセスパーティション番号２００ｂ４、プロセス番号２００ｂ１、論理アドレス２００ｂ２が伝えられ、該当するアドレスのデータ２００ｂ３が読み出される。

【００４３】

図１２にアドレス変換手段２００内部にあるアダプタＴＬＢ２０００の構成を示す。アダプタＴＬＢ２０００では、パーティション番号２００１、プロセス番号２００２、論理ページ番号２００３から、物理ページ番号２００４にアドレス変換を行うことが可能である。アドレス変換方式や、ＴＬＢの構成方式について

は公知の技術であるので説明を省略する。

【００４４】

次いで、送信スケジューラ１４０の処理を図３に基づき詳細に述べる。以下の説明では、ローカルなレジスタｐは次に送信するべきパーティション番号を記憶するレジスタであり、ｎは現在送信中のパーティションにおいて連続して送信したパケット数である。

【００４５】

まず、ｐは０に初期化される（ステップ５１００）。各パーティションに対応するデータの送信に先立ち、ｎが０にリセットされる（ステップ５１０１）。通信先のパーティションｐに対応する受信バッファが満杯（フル）でないことが確認（ステップ５１０７）される。フルの場合はステップ５１０６に飛び、次のパーティションの処理を行う。受信バッファが満杯（フル）でないことが確認されると、パーティションｐのメッセージを１パケット分送信することを指示し（ステップ５１０２）、ｎがインクリメントされる（ステップ５１０３）。

【００４６】

ここで、ｎがパーティション毎にあらかじめ決められている、連続して送信できるパケット数の上限に達したかが検査（ステップ５１０４）され、上限に達していない場合はステップ５１０２に戻り、該当するパーティションのデータの送信を続ける。ｎがパケット数の上限に達した場合には、送信回路１３０に信号１４０ａを通じて送信中断１４０ａ２を指示し（ステップ５１０５）、ｐをインクリメントし、ステップ５１０１に戻り、次のパーティションのデータ送信を開始する（ステップ５１０６）。ｐがパーティション数を超えた場合には、０にラップアラウンドする。

【００４７】

ここで、連続して送信できるパケット数の上限は、送受信割当てレジスタ１５０の内部にパーティション毎に記憶されている（図４）。例えば図４の場合、送信回路は、

・パーティション０の通信を３パケット送信

・パーティション１の通信を１パケット送信

という動作を繰り返すことにより、パーティション０は全通信量の７５％を占有し、パーティション１は２５％を占有することになる。

【００４８】

以上のように、パーティション制御プログラムが送受信割当てレジスタ１５０にあらかじめ適当な値を設定しておけば、Ｉ／Ｏアダプタ１００のハードウェアが送受信の割当てを動的に実行する。さらに、パーティションへのＩ／Ｏ割当量を変更する場合には、パーティション制御プログラムが送受信割当てレジスタ１５０の値を変更することにより、パーティション上で動作するＯＳやプログラムに対して透過的（パーティション上のプログラムやＯＳに介入することなく、パーティション制御プログラムの処理だけで）に実現できる。

【００４９】

次いで受信側の動作を述べる。図１０に受信回路１８０の詳細、図１１に受信側ＤＭＡＣ１８３０の動作フロー、図５に受信スケジューラ１９０の動作フローを示す。受信側の動作も送信側と基本的に同一であるので、詳細な説明は略す。相違点は、送信側ＤＭＡＣ１３３０はデータを主記憶３０から読み出して、信号１３０ａを通じてネットワークへデータを出力するのに対し、受信ＤＭＡＣ１８３０はデータを受信バッファ１７０から読込み、信号１８０ａ（１８０ａ３）を通じてデータを主記憶３０に書込むことである。また、相手側の送信バッファフルチェックに相当する処理は存在しない。

【００５０】

図１４に受信バッファ１７０の構造を記す。受信バッファ１７０は各パーティション毎に独立したバッファ１７００～１７０３を持っており、ネットワークパケットのヘッダにあるパーティション番号４３３０等に対応したバッファにデータを貯える。受信バッファ１７０は、ネットワークから各パーティションのパケットが入力される順番と、アダプタのスケジューラが実際にデータを主記憶に書込む順番の差を吸収するために設けられている。スケジューリングされたパーティションの入力バッファが空であった場合には、データの入力が行われない。あるパーティションの入力バッファがフルになった場合には、ネットワークのフロー制御により、該当するパーティションに対する通信は抑止される。

【００５１】

以上の説明はパーティションがＩ／Ｏアダプタを共有している場合に、各パーティションへのＩ／Ｏ割当てを時分割で制御する機構を中心に説明した（図２（ａ）に該当）。次に、システム内に複数のＩ／Ｏアダプタが存在し、各パーティションに必要な個数のＩ／Ｏアダプタを割り振ることにより、各パーティションのＩ／Ｏ割当てを空間分割で制御する方式を述べる（図２（２）に該当）。パーティションへのＩ／Ｏアダプタの割当ては、アクセスパーティション判定回路１１０において、上記Ｉ／Ｏアダプタを割当てるパーティションに対応するｖａｌｉｄビットを１にすることにより実現できる。

【００５２】

ここで、各パーティションへのＩ／Ｏ割当てを変更する場合、時分割方式では、パーティション上で動作しているＯＳに透過的にＩ／Ｏの割当てを変更できるのに対して、空間分割方式ではパーティション（ＯＳ）に割り当てられるＩ／Ｏアダプタの数が変わるため、パーティション制御プログラムとパーティション上で動作しているＯＳが協調動作してパーティションへのＩ／Ｏの割当てを変更する必要がある。

【００５３】

図１６にＩ／Ｏアダプタのパーティション間での移動アルゴリズムを示す。この場合、Ｉ／Ｏ割当てを変更するパーティションで動作しているＯＳがＩ／Ｏアダプタの動的なホットプラグに対応していることが大前提である。あるＩ／Ｏアダプタを現在Ｉ／Ｏアダプタを所有しているパーティション（旧パーティション）から別のパーティション（新パーティション）に移動させる場合、以下のステップが必要である（図１６）。

【００５４】

まず、パーティション制御プログラムは、旧パーティションのＯＳに上記Ｉ／Ｏアダプタの使用停止を指示する（ステップ６０００）。旧パーティションのＯＳは、上記Ｉ／Ｏアダプタの使用を停止し、ＯＳから切り離す（ステップ６００１）。続いて、パーティション制御プログラムは、Ｉ／Ｏアダプタのアクセスパーティション判定回路１１０にある、ベースアドレスレジスタ１１００のｖａｌ

ｉｄビット（１１１０～１１１３）等を書き換え、上記Ｉ／Ｏアダプタへの旧パーティションからのアクセスを禁止し、新パーティションからのアクセスを許可する（Ｉ／Ｏアドレスの再割当てが必要な場合は、再割当てを行う）（ステップ６００２）。次いで新パーテシションのＯＳに上記Ｉ／Ｏアダプタの使用開始を指示し（ステップ６００３）、新パーティションのＯＳが上記Ｉ／Ｏアダプタを使用開始する（ステップ６００４）。上記の方式により、Ｉ／Ｏアダプタのロジカルなホットプラグ、パーティション間の移動を実現し、空間分割によるパーティションのＩ／Ｏ割当ての動的な制御を可能にする。

【００５５】

パーティション毎のＩ／Ｏ割当て制御を行うためには、パーティション毎にＩ／Ｏ性能を正確に知る手段が必要である。上記の目的のために、各Ｉ／Ｏアダプタにはモニタリング手段２１０を設ける。

【００５６】

図１５にモニタリング手段２１０の内部構成を示す。各パーティション、送信／受信に対応して、性能モニタリングカウンタ（２１００～２１０３、２１１０～２１１３）が用意されており、パーティション別にＩ／Ｏ性能を計測することを可能にする。計測する項目としては以下の項目が挙げられる。

・送信側

送信までの待ち時間（送信がエンキューされてから実際にネットワークに送信されるまでの待ち時間）

送信データ量（時間で割ることにより、送信スループットが求まる）

・受信側

受信までの待ち時間（ネットワークから受信パケットが着いてから、実際にデータが主記憶にコピーされるまでの待ち時間）

受信データ量（時間で割ることにより、受信スループットが求まる）

以上の項目をモニタリングすることにより、各パーティションのＩ／Ｏ待ち時間と、Ｉ／Ｏスループットが求まり、性能ボトルネックを知ることができる。モニタリングの詳細は従来技術であるので説明を略す。

【００５７】

さらに、以下のような稼働方法と機能を持つことにより、上記で述べた、パーティション毎のＩ／Ｏ割当ての制御をユーザへの課金に活用することができる。

（１）あらかじめ、ユーザと、パーティションへの標準のＩ／Ｏ割当量、パーティションのＩ／Ｏ割当てを変更する条件、パーティションのＩ／Ｏ割当てを増やした場合に払うべき割当て使用料について契約を結ぶ。

（２）パーティション制御プログラムが、あるパーティションのＩ／Ｏ割当てを変更した場合には、Ｉ／Ｏ割当てを変更した時間、変更後のＩ／Ｏ割当量をログに記録する。

（３）ユーザの使用料を計算する際に、前記ログを参照し、パーティションの標準Ｉ／Ｏ割当量よりＩ／Ｏ割当てが増やされていた時間に応じて、あらかじめ決めた契約に従い、ユーザへの課金を割り増す。

【００５８】

以上の手段により、本Ｉ／Ｏ割当て制御機構によりユーザへのＩ／Ｏ割当てが増加された時間に応じて、割増し課金を賦課することができる。

【００５９】

以上では、パーティションのＩ／Ｏ割当てを制御する機構の詳細を中心に説明した。以下では図１７～図２０を用いて、上記の機構を使用してどのようにシステム性能を制御するかについて、いくつかの場合に分けて詳細に説明する。また、以下ではＩ／Ｏを時分割で割当てる場合について説明を行うが、空間分割でも同様に制御することができる。

（１）オペレータによるＩ／Ｏ割当ての手動変更

図１７にオペレータがパーティション制御プログラムに対して、パーティションのＩ／Ｏ割当て変更を指示するインタフェースを示す。図で７０００はコンソールの画面である。上半分７０１０では、各パーティションのＩ／Ｏ性能モニタ結果を示し、下半分７０２０はパーティションのＩ／Ｏ割当て量の変更を指示する画面である。７０２０では各パーティションの旧Ｉ／Ｏ割当量を示し、入力領域７０３０～７０３１で新しいパーティションのＩ／Ｏ割当量を指示する。図ではパーティション０とパーティション１のＩ／Ｏ割当て量を５０％、５０％から、７５％、２５％への変更を指示している。

【００６０】

パーティション制御プログラムでは、上記のパーティション毎のＩ／Ｏ割当ての割合をなるべく簡単な（場合によっては近似の）整数比に直し、図１の送受信割当てレジスタ１５０に各パーティションが連続して送受信できるパケット数を指示する。図１７の場合、７５％：２５％＝３：１であるので、送受信割当てレジスタのパーティション０に該当する領域１５００（図４）には３が、パーティション１に該当する領域１５０１には１が書込まれる。

【００６１】

上記の手段により、パーティション０とパーティション１のＩ／Ｏ割当ては７５％と２５％に制御することができる。以上の手段により、オペレータがシステムの稼動状況をモニタリングして、Ｉ／Ｏ割当てを変更することができる。

【００６２】

図１７では送信ディレイをモニタしているが、受信ディレイをモニタする場合も同様である。

（２）設定ファイルによるＩ／Ｏ割当ての予約、自動変更

図１８に、各パーティションのＩ／Ｏ割当てを設定するファイルの一例を示す。図では毎日８時と１８時にパーティションのＩ／Ｏ割当ての設定をつぎのように変更することを指示している。

・午前８時にパーティション０、パーティション１のＩ／Ｏ割当てを５０％、５０％に設定する。

・午後１８時にパーティション０、パーティション１のＩ／Ｏ割当てを７５％、２５％に設定する。

【００６３】

上記の設定ファイルに基づき、パーティション制御プログラムは指定された時間に、送受信割当てレジスタの内容を変更する。具体的な変更方法は（１）と同様である。

（３）ＣＰＵ割当てへの自動追従

パーティション制御プログラムが各パーティションへのＣＰＵ割当てを変更する場合、Ｉ／Ｏ割当てを自動的にＣＰＵ割当てと同じ割合で変更する。Ｉ／Ｏ割

当てだけを積極的に変更する要求がない場合に有効な方法である。

（４）Ｉ／Ｏ割当ての自動変更

各パーティションのＩ／Ｏ性能計測結果、および、あらかじめ決められたＩ／Ｏ割当て変更条件に基づき、パーティション制御プログラムが、オペレータの介在無しにＩ／Ｏ割当てを変更する。

【００６４】

図１９にＩ／Ｏ割当ての自動変更方式を示す。パーティション制御プログラムは定期的に図１９に示す処理を実行する。まず、各パーティションのＩ／Ｏ性能をモニタリングする（ステップ７１００）。モニタリングした結果、あらかじめ決められていたＩ／Ｏ割当て変更条件に合致するかどうかがチェックされる（ステップ７１０１）。ここで、Ｉ／Ｏ割当て変更条件は、次のような条件とアクションの組で表される。

・条件……パーティション０のＩ／Ｏディレイが５００ｍｓ以上

・アクション……パーティション０のＩ／Ｏ割当て７５％に増加

条件が満たされた場合、指定するアクションが実行され、Ｉ／Ｏ割当て量が変更される（ステップ７１０２）。

【００６５】

さらに、上記パーティションのＩ／Ｏ割当てが増加されたことをログに記録する（ステップ７１０３）。これにより、上記パーティションを使用するユーザからＩ／Ｏ割当てを増加した時間に応じた割増し使用量を賦課することができる。

（５）ＳＬＡ保証

上記（４）で述べたＩ／Ｏ割当ての自動保証を発展させ、Ｉ／Ｏ割当てを変更することにより、ユーザプログラムのＳＬＡを保証することが可能である。インタネットデータセンタ等ではサーバのコストを下げるために１台のサーバに複数のパーティションを設け、複数のユーザのプログラムを実行させる事が望まれる。その場合、ユーザと約束した応答時間等のＳＬＡを守る事が求められる。刻々と変わるインタネットの負荷に対応するためには、ＣＰＵ割当てのみならずＩ／Ｏ割当てを負荷に合わせてアダプティブに変更する必要がある。

【００６６】

パーティション制御プログラムはＳＬＡを保証するために、図２０に示す処理を実行する。まず、各パーティションのユーザプログラムの性能を測定（ステップ７２０１）し、ＳＬＡが満たされるかを判断する（ステップ７２０２）。ＳＬＡが満たされていない場合（もしくは、ＳＬＡが未達になる可能性が強い場合）、ＳＬＡ未達の原因を知るために、ＣＰＵ性能、Ｉ／Ｏ性能をモニタリングする（ステップ７２０３）。具体的には、該当するパーティションへのＣＰＵおよびＩ／Ｏ割当てと、ＣＰＵ、Ｉ／Ｏの使用時間から、上記パーティションのＣＰＵ使用率およびＩ／Ｏ使用率を測定し、ＣＰＵネックおよびＩ／Ｏネックのどちらであるかを判定（ステップ７２０４）する。例えば、ＣＰＵ使用率３０％、Ｉ／Ｏ使用率９５％の場合は、Ｉ／Ｏネックであると判定する。

【００６７】

Ｉ／Ｏネックである場合には、他のパーティションのＩ／Ｏ使用率に余裕があるかどうかを判定し（ステップ７２０７）、余裕がある場合にはシステムのＩ／Ｏ割当を調整し、性能ネックになっているパーティションのＩ／Ｏ割当を増加させる（ステップ７２０８）。ＣＰＵネックであると判定された場合も同様の処理を行う（ステップ７２０５、７２０６）。他のパーティションのＣＰＵまたはＩ／Ｏ使用量に余裕がなく、必要なリソースの割当てを増やすことができなかった場合には、ＳＬＡを守ることができないことがログに記録される（７２０９）。この記録は、後刻ユーザに対して、ＳＬＡを守れなかった時間に対して返金または課金を割り引く処理などに使用される。

【００６８】

以上述べた構成により、複数のパーティションを持ったシステムにおいて、各パーティションのＩ／Ｏの割当てを動的に変更する事を可能にするとともに、上記機構を用いてパーティションで実行しているユーザプログラムのＳＬＡ保持を実現する。

＜変形例＞

本発明は以上の実施の形態に限定されず、いろいろな変形例に適用可能である。（１）例えば、以上の実施の形態においては、アクセスパーティション判定回路１１０において、アクセスされたアドレスからアクセスされたパーティション

を判断していたが、

・チップセットからＩ／Ｏアダプタへの信号１１０ｂにパーティション番号を示すビットを追加する、

・アダプタ側のアクセスパーティション判定回路１１０の内部に、アクセスパーティション番号を明にセットするレジスタを設ける（この場合、Ｉ／Ｏを起動する前に、ＯＳまたはパーティション制御プログラムが上記レジスタをセットする）、

のように、明示的にパーティションを指示する方式も可能である。

（２）また、以上の実施の形態において、送・受信割当てレジスタ（１５０）では、送信、受信に共通に上限パケット数を記憶していたが、送信用と受信用別々に上限パケット数を記憶することにより、送信、受信のＩ／Ｏ割当てを別個に制御することが可能である。

（３）また、以上の実施の形態においては、各パーティションの入出力をパケット毎に切り替えていたが、タイムスライスにより、一定時間毎に切り替えることも可能である。その場合、送受信ネットワークにおいて、パーティション毎のバーチャル・チャネル制御が必要になる。

（４）また、以上の実施の形態においては、送受信割当てレジスタ１５０は各パーティションが連続して送受信できるパケット数という、単純なパラメータを記憶しているが、送信スケジューラ１４０等に機能を追加することにより、より複雑な制御を行うことが可能である。

【００６９】

例えば、あるパーティションが連続して送信できるサイクル数を分数で記憶し、図３において、複数の繰り返しで１回該当するパーティションのデータ送受信を行う、等の制御が可能である。

（５）また、以上の実施の形態においては、送受信のスケジューリングをハードウェア１４０、１９０、１５０で行わせていたが、１パケット送信する毎にハイパバイザやＳＶＰ等のパーティション制御プログラムに割込み、パーティション制御プログラムがソフト的にＩ／Ｏのスケジューリングを行う方式も可能である。

（６）また、以上の実施の形態においては、ＳＬＡに基づき、パーティションへのＩ／Ｏ割当てを判断する主体が、パーティション制御プログラムであったが、パーティション制御プログラムの外部のプログラム（ポリシ制御プログラム）でＩ／Ｏ割当て条件を判断し、パーティション制御プログラムに実際のＩ／Ｏ割当ての変更を依頼する方法をとることも可能である。さらに、該当する計算機の外部の計算機（ポリシサーバ等）でシステム全体のＳＬＡを判断し、パーティション制御プログラムにＩ／Ｏ割当ての変更を依頼する方法も可能である。

【００７０】

上記の場合、外部のサーバからパーティション制御プログラムに対しＩ／Ｏの割当ての変更を指示するために、システム内の各パーティションに対するＩ／Ｏ割当て量のリストを与えるＡＰＩが必要になる。上記のＡＰＩを用いることにより、外部のプログラムから強制的にシステムの各パーティションのＩ／Ｏ割当てを変更することが可能になる。

【００７１】

【発明の効果】

本発明によれば、パーティションに分けられた計算機において、各パーティションに割当てられるＩ／Ｏ能力を、時分割、空間分割等の方法によって制御する手段を、Ｉ／Ｏアダプタとパーティション制御プログラムに持たせる。それにより、各パーティションへのＩ／Ｏ割当てをＣＰＵ割当てとは独立に制御するとともに、動的に変更することを可能にする。さらに、Ｉ／Ｏ割当ての動的変更により、パーティション上のユーザプログラムのＳＬＡ保持を実現する。

【図面の簡単な説明】

【図１】

本発明の一実施の形態を示す計算機のブロック図。

【図２】

Ｉ／Ｏ割当ての例を示す説明図。

【図３】

本発明の一実施の形態における送信スケジューラの動作フロー図。

【図４】

本発明の一実施の形態における送受信割当てレジスタの説明図。

【図５】

本発明の一実施の形態における受信スケジューラの動作フロー図。

【図６】

本発明の一実施の形態におけるアクセスパーティション判定回路のブロック図。

【図７】

本発明の一実施の形態におけるコントロールレジスタ群を示すブロック図。

【図８】

本発明の一実施の形態における送信回路のブロック図。

【図９】

本発明の一実施の形態における送信側ＤＭＡＣの動作フロー図。

【図１０】

本発明の一実施の形態における受信回路のブロック図。

【図１１】

本発明の一実施の形態における受信側ＤＭＡＣの動作フロー図。

【図１２】

本発明の一実施の形態におけるアダプタＴＬＢの構成例を示す説明図。

【図１３】

本発明の一実施の形態におけるＩ／Ｏパケットのフォーマット例を示す説明。

【図１４】

本発明の一実施の形態における受信バッファの説明図。

【図１５】

本発明の一実施の形態におけるモニタリング手段の説明図。

【図１６】

本発明の一実施の形態における空間分割しているＩ／Ｏアダプタをパーティション間で移動する場合の処理フロー図。

【図１７】

本発明の一実施の形態におけるコンソールからＩ／Ｏ割当てを変更する場合の

コンソール画面の説明図。

【図１８】

本発明の一実施の形態における設定ファイルによりＩ／Ｏ割当てを予約する場合の設定ファイルの構成例を示す説明図。

【図１９】

本発明の一実施の形態におけるＩ／Ｏ割当てを自動変更する処理フロー図。

【図２０】

本発明の一実施の形態におけるＳＬＡを保証するための処理フロー図。

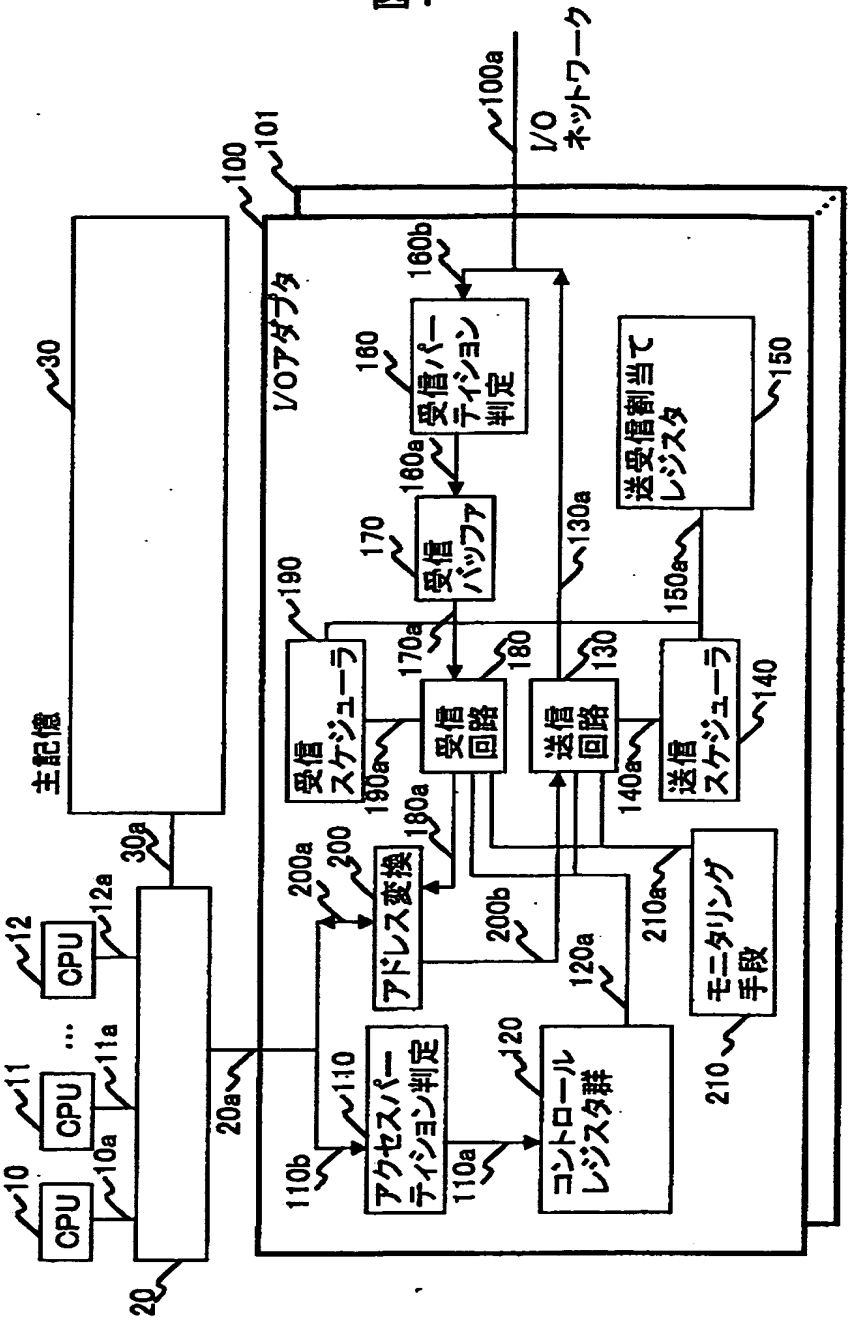
【符号の説明】

１０…ＣＰＵ、１１…ＣＰＵ、１２…ＣＰＵ、２０…チップセット、３０…主記憶、１００…Ｉ／Ｏアダプタ、１０１…Ｉ／Ｏアダプタ、１００ａ…外部Ｉ／Ｏネットワーク。

【書類名】図面

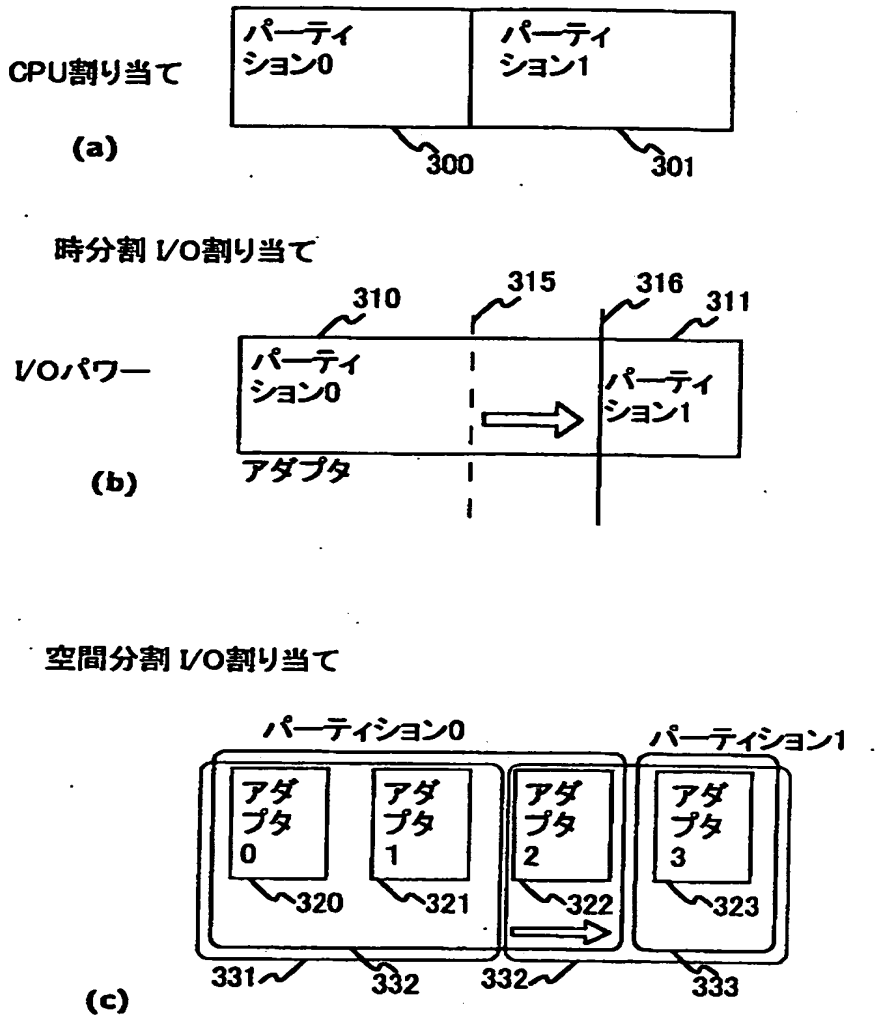
【図１】

図１


I/O ネットワーク
100a
101
100
I/Oアダプタ
160b
160
受信パーティション判定
160a
170
受信バッファ
190
受信スケジューラ
190a
170a
180
受信回路
180a
130a
130
送信回路
140a
140
送信スケジューラ
150
送受信割当てレジスタ
150a
30
主記憶
30a
200a
200
アドレス変換
200b
120a
120
コントロールレジスタ群
210a
210
モニタリング手段
110
アクセスパーティション判定
110a
110b
20a
20
12
CPU
12a
11
CPU
11a
10
CPU
10a

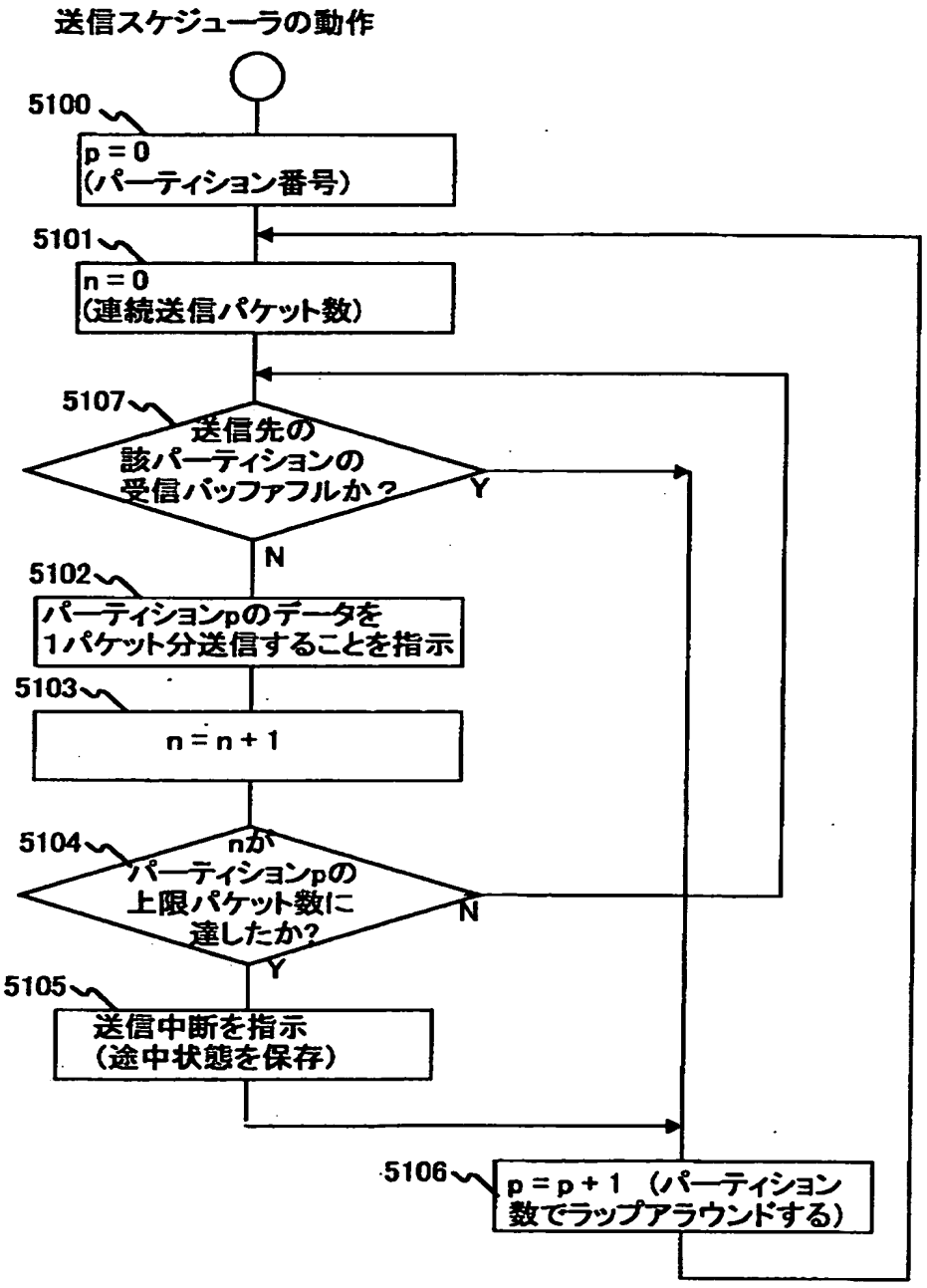
【図２】

図2


CPU割り当て
パーティション0
パーティション1
(a)
300
301
時分割 I/O割り当て
310
315
316
311
I/Oパワー
パーティション0
パーティション1
(b)
アダプタ
空間分割 I/O割り当て
パーティション0
パーティション1
アダプタ0
アダプタ1
アダプタ2
アダプタ3
320
321
322
323
331
332
332
333
(c)

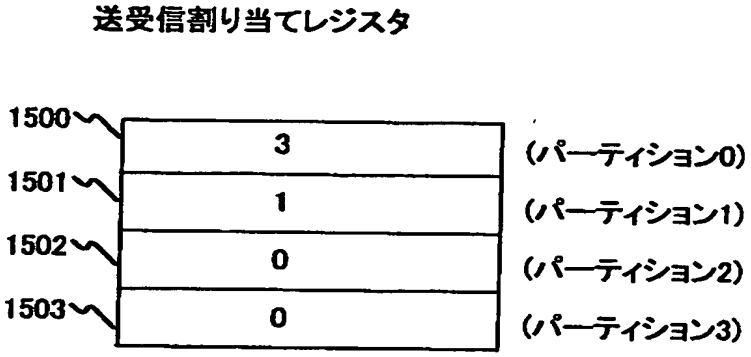
【図３】

図3


送信スケジューラの動作
5100
p = 0
(パーティション番号)
5101
n = 0
(連続送信パケット数)
5107
送信先の
該パーティションの
受信バッファフルか？
Y
N
5102
パーティションpのデータを
1パケット分送信することを指示
5103
n = n + 1
5104
nが
パーティションpの
上限パケット数に
達したか?
N
Y
5105
送信中断を指示
(途中状態を保存)
5106
p = p + 1 (パーティション
数でラップアラウンドする)

【図４】

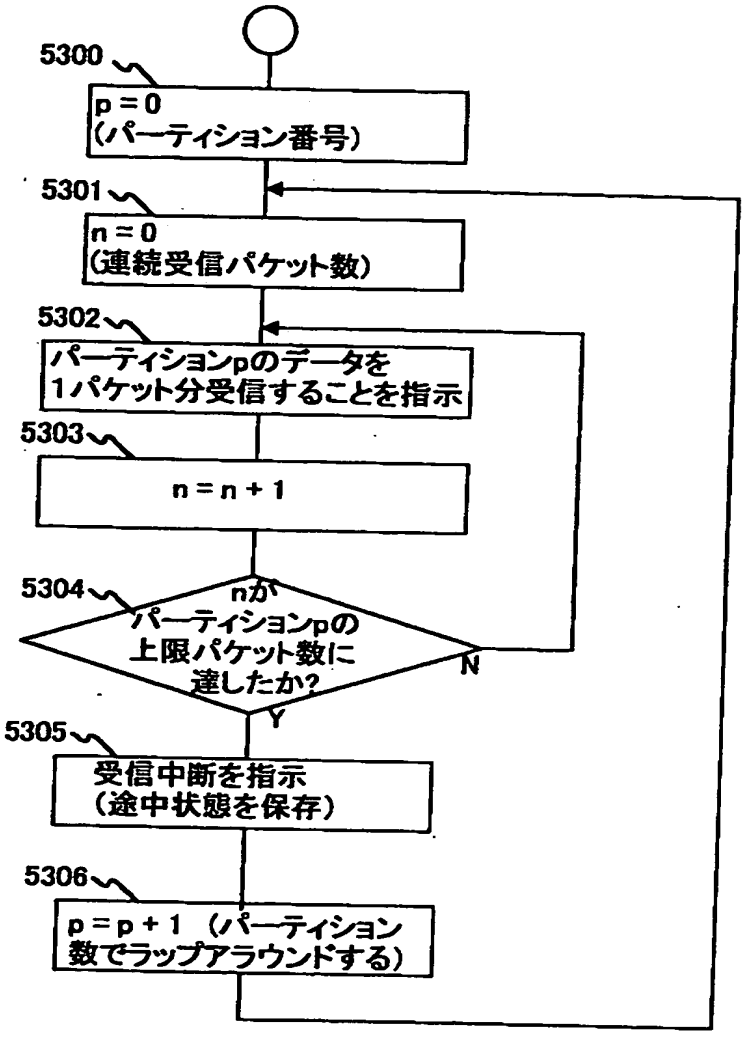
図4


送受信割り当てレジスタ
1500
3
(パーティション0)
1501
1
(パーティション1)
1502
0
(パーティション2)
1503
0
(パーティション3)

【図５】

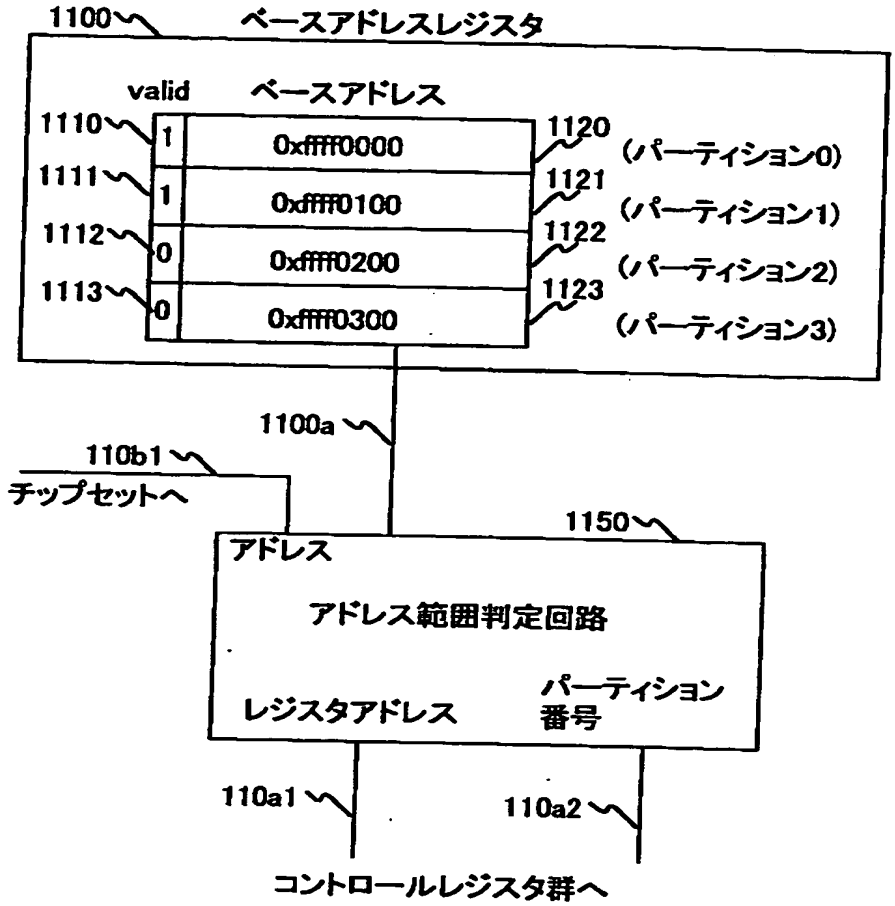
図5

受信スケジューラの動作



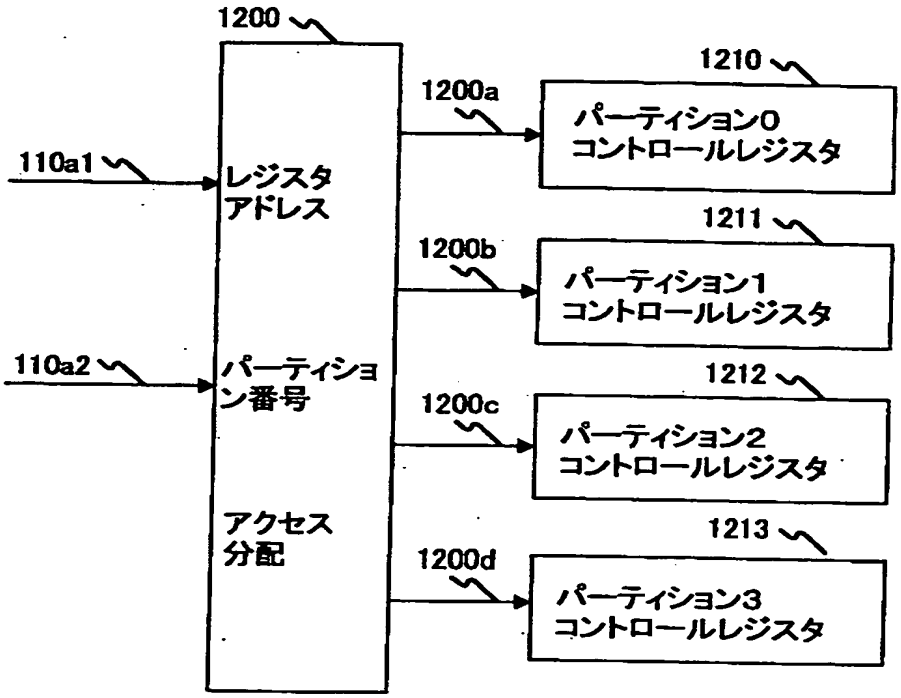
【図６】

図6


1100
ベースアドレスレジスタ
valid
ベースアドレス
1110
1
0xffff0000
1120
(パーティション0)
1111
1
0xffff0100
1121
(パーティション1)
1112
0
0xffff0200
1122
(パーティション2)
1113
0
0xffff0300
1123
(パーティション3)
1100a
110b1
チップセットへ
1150
アドレス
アドレス範囲判定回路
レジスタアドレス
パーティション番号
110a1
110a2
コントロールレジスタ群へ

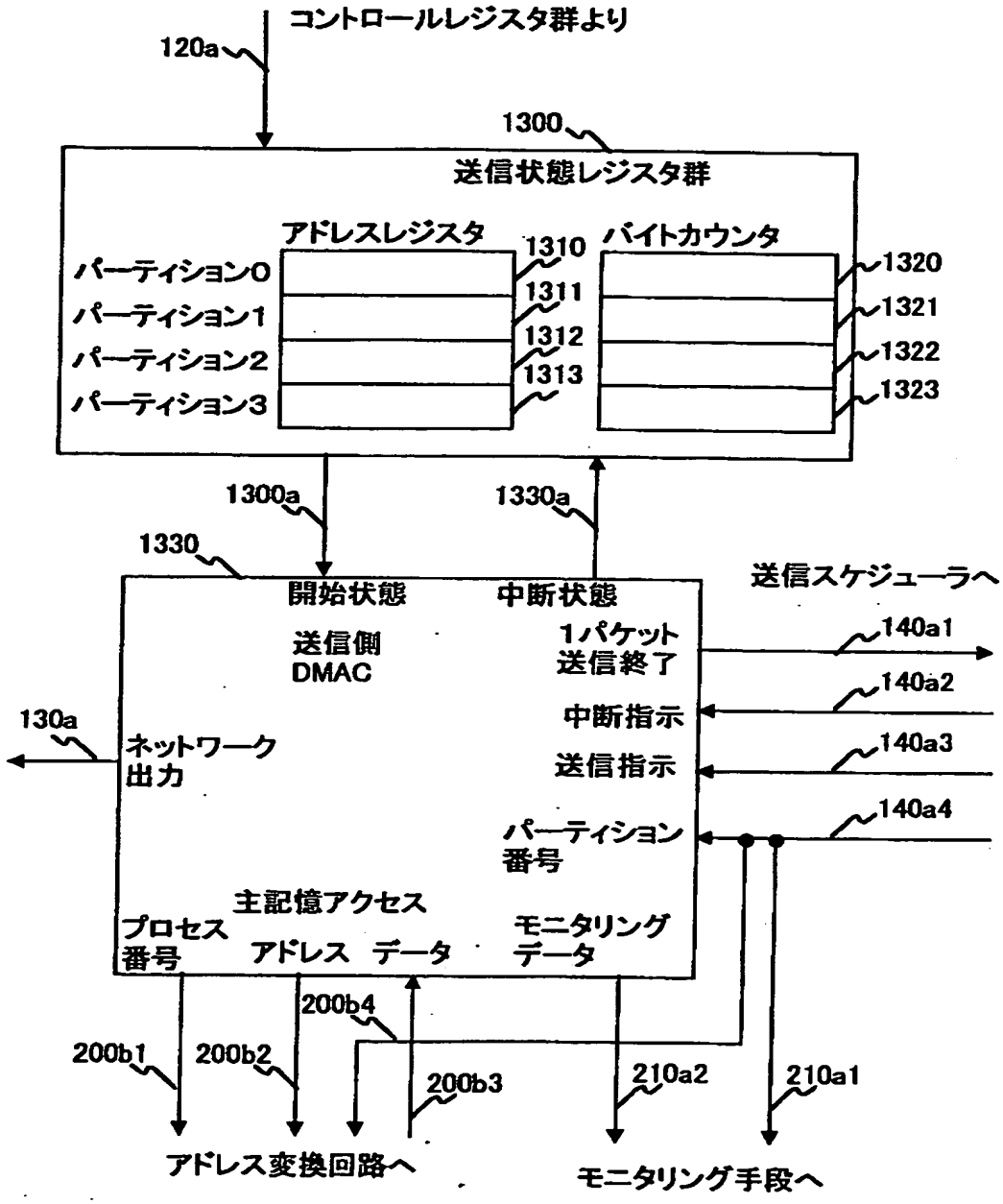
【図７】

図７


1200
1200a
1210
パーティション0
コントロールレジスタ
110a1
レジスタ
アドレス
1211
1200b
パーティション1
コントロールレジスタ
110a2
パーティショ
ン番号
1212
1200c
パーティション2
コントロールレジスタ
アクセス
分配
1213
1200d
パーティション3
コントロールレジスタ

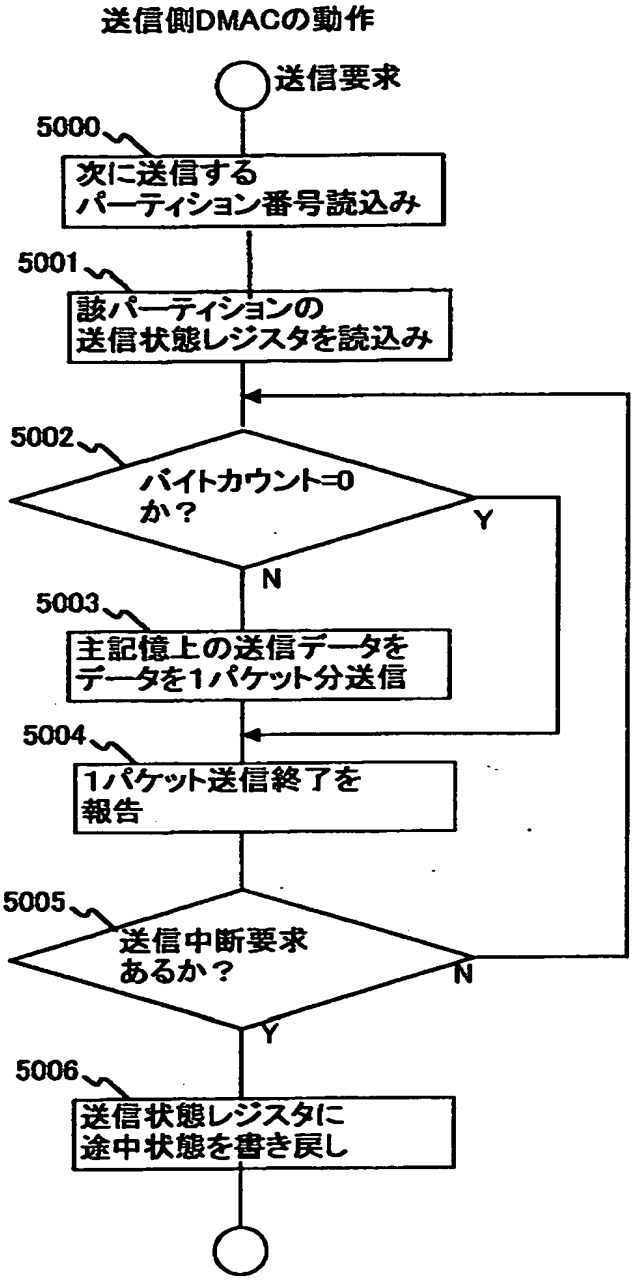
【図８】

図8


コントロールレジスタ群より
120a
1300
送信状態レジスタ群
アドレスレジスタ
1310
バイトカウンタ
パーティション0
パーティション1
パーティション2
パーティション3
1311
1312
1313
1320
1321
1322
1323
1300a
1330a
1330
開始状態
中断状態
送信スケジューラへ
送信側
DMAC
1パケット
送信終了
140a1
140a2
中断指示
130a
ネットワーク
出力
140a3
送信指示
140a4
パーティション
番号
主記憶アクセス
プロセス
番号
アドレス
データ
モニタリング
データ
200b4
200b1
200b2
200b3
210a2
210a1
アドレス変換回路へ
モニタリング手段へ

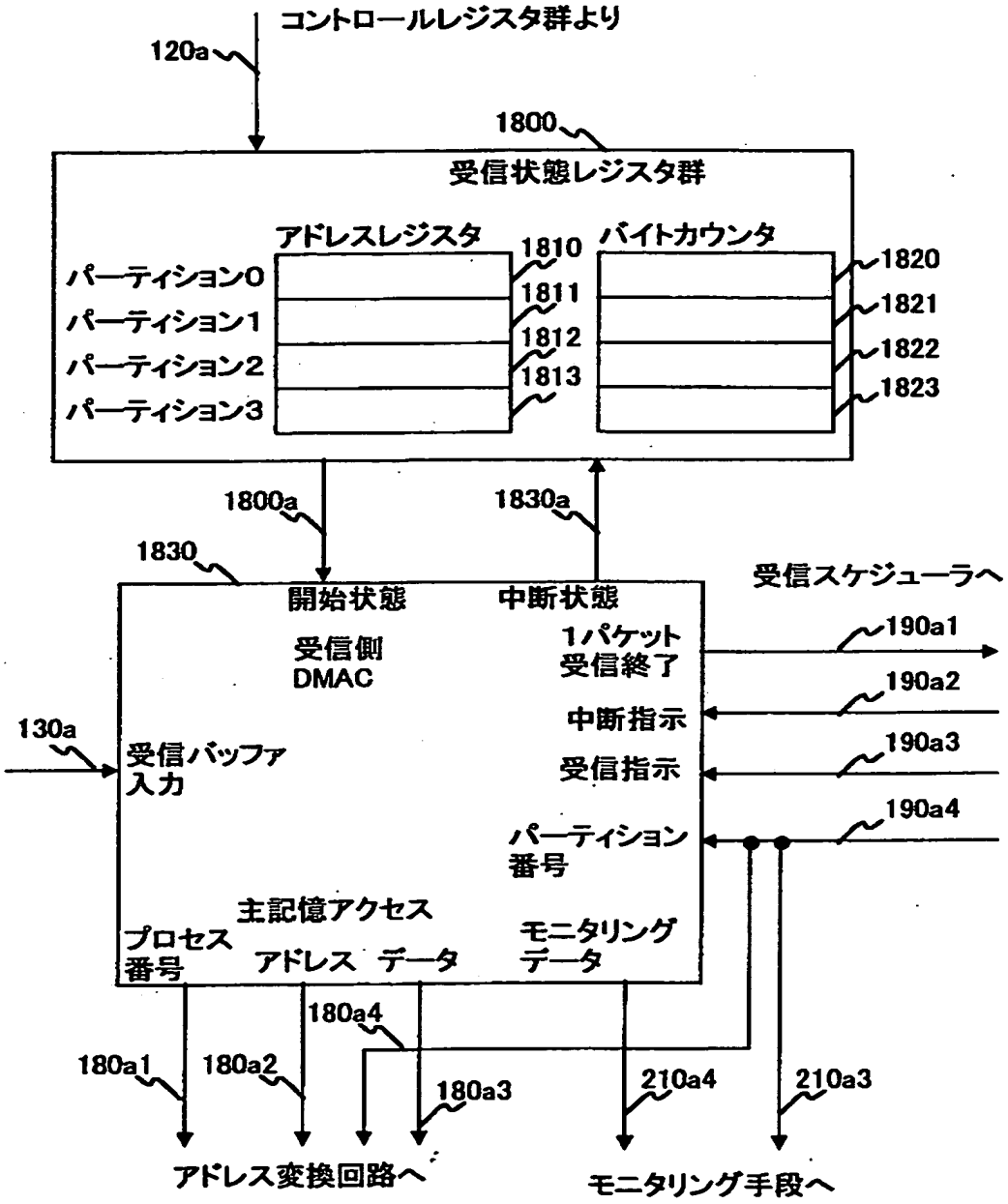
【図９】

図9


送信側DMACの動作
送信要求
5000
次に送信する
パーティション番号読込み
5001
該パーティションの
送信状態レジスタを読込み
5002
バイトカウント=0
か？
Y
N
5003
主記憶上の送信データを
データを1パケット分送信
5004
1パケット送信終了を
報告
5005
送信中断要求
あるか？
N
Y
5006
送信状態レジスタに
途中状態を書き戻し

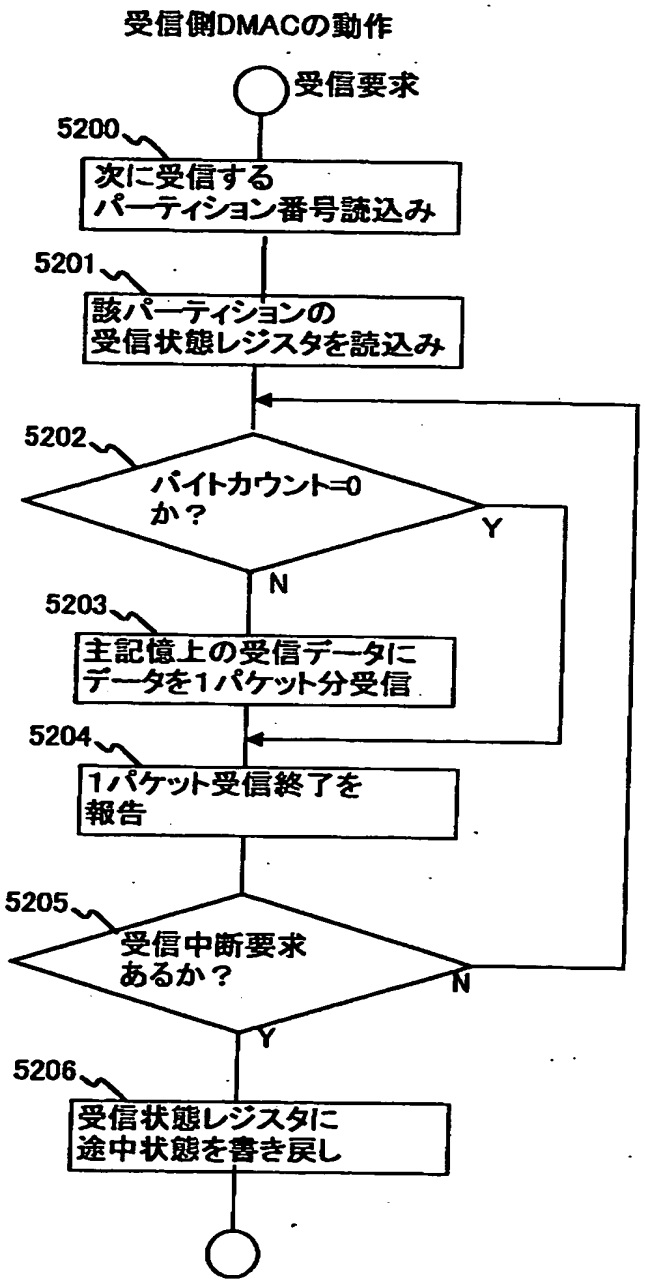
【図１０】

図10


コントロールレジスタ群より
120a
1800
受信状態レジスタ群
アドレスレジスタ
1810
バイトカウンタ
パーティション0
パーティション1
パーティション2
パーティション3
1811
1812
1813
1820
1821
1822
1823
1800a
1830a
1830
開始状態
中断状態
受信スケジューラへ
受信側
DMAC
1パケット
受信終了
190a1
190a2
中断指示
130a
受信バッファ
入力
受信指示
190a3
190a4
パーティション
番号
主記憶アクセス
プロセス
番号
アドレス
データ
モニタリング
データ
180a4
180a1
180a2
180a3
210a4
210a3
アドレス変換回路へ
モニタリング手段へ

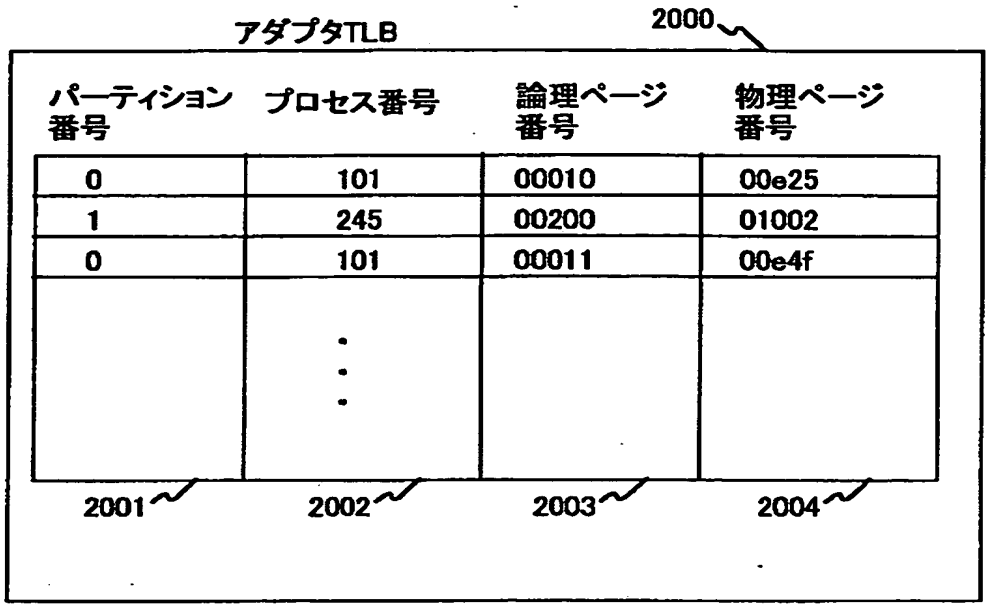
【図１１】

図11


受信側DMACの動作
受信要求
5200
次に受信する
パーティション番号読込み
5201
該パーティションの
受信状態レジスタを読込み
5202
バイトカウント=0
か？
Y
N
5203
主記憶上の受信データに
データを1パケット分受信
5204
1パケット受信終了を
報告
5205
受信中断要求
あるか？
N
Y
5206
受信状態レジスタに
途中状態を書き戻し

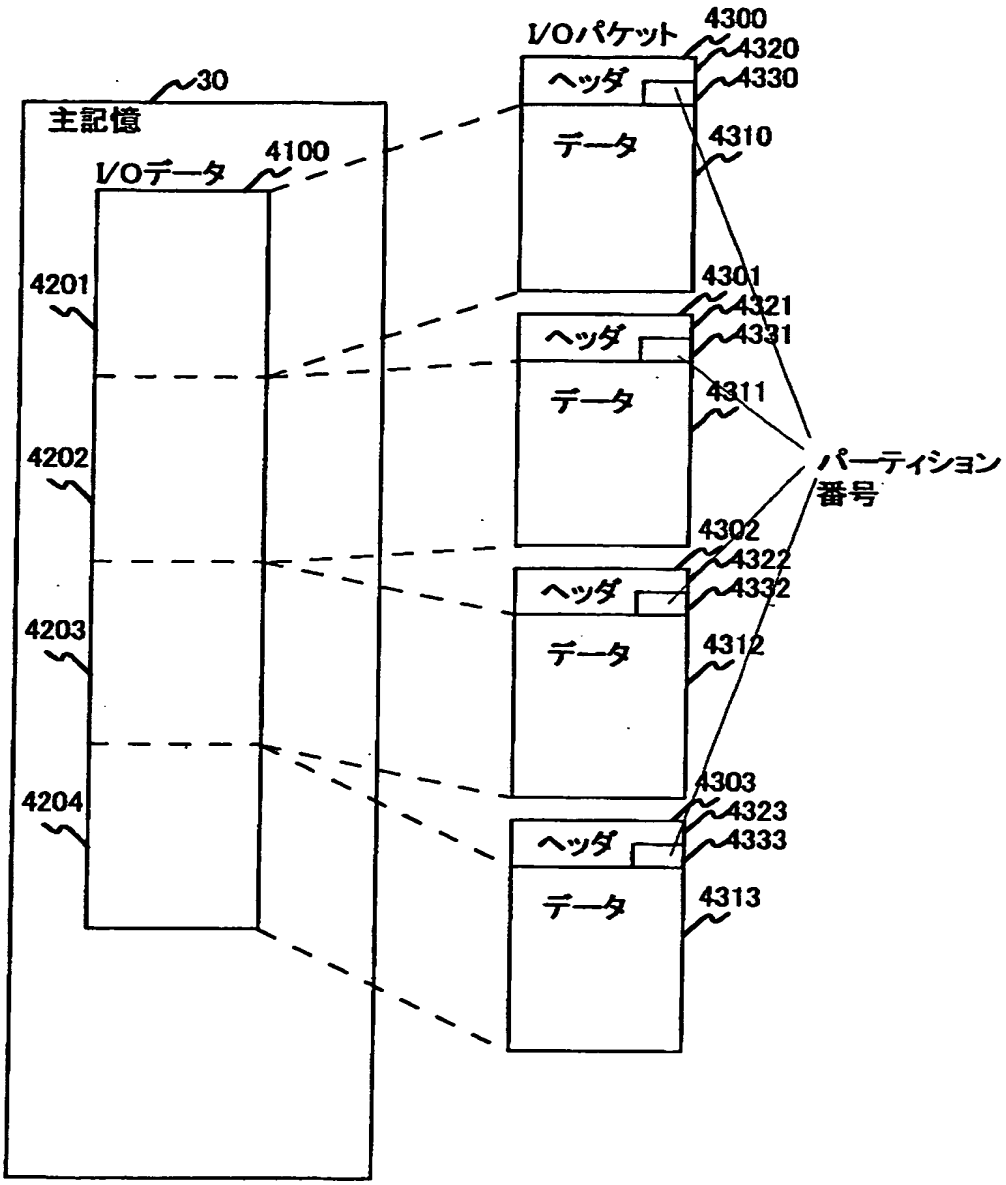
【図１２】

図12


アダプタTLB
2000
パーティション番号
プロセス番号
論理ページ番号
物理ページ番号
0
101
00010
00e25
1
245
00200
01002
0
101
00011
00e4f
2001
2002
2003
2004

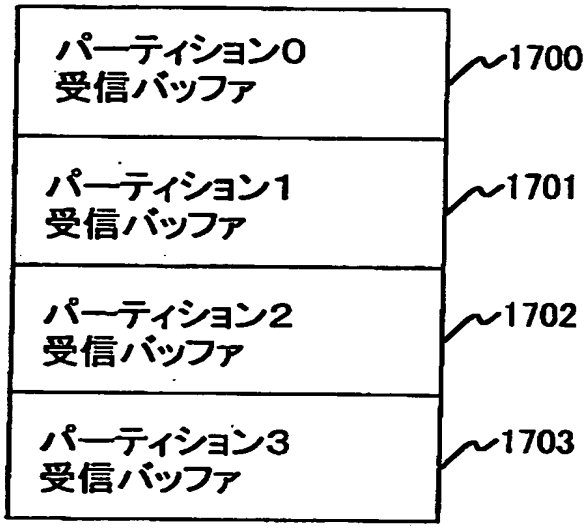
【図１３】

図13


30
主記憶
I/Oデータ
4100
4201
4202
4203
4204
I/Oパケット
4300
ヘッダ
4320
4330
データ
4310
4301
4321
4331
4311
4302
4322
4332
4312
4303
4323
4333
4313
パーティション番号

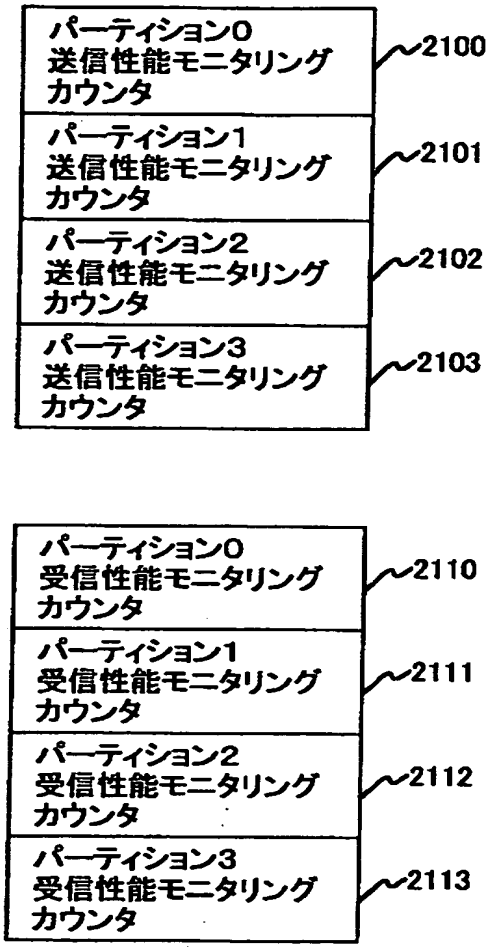
【図１４】

図14


パーティション0
受信バッファ
1700
パーティション1
受信バッファ
1701
パーティション2
受信バッファ
1702
パーティション3
受信バッファ
1703

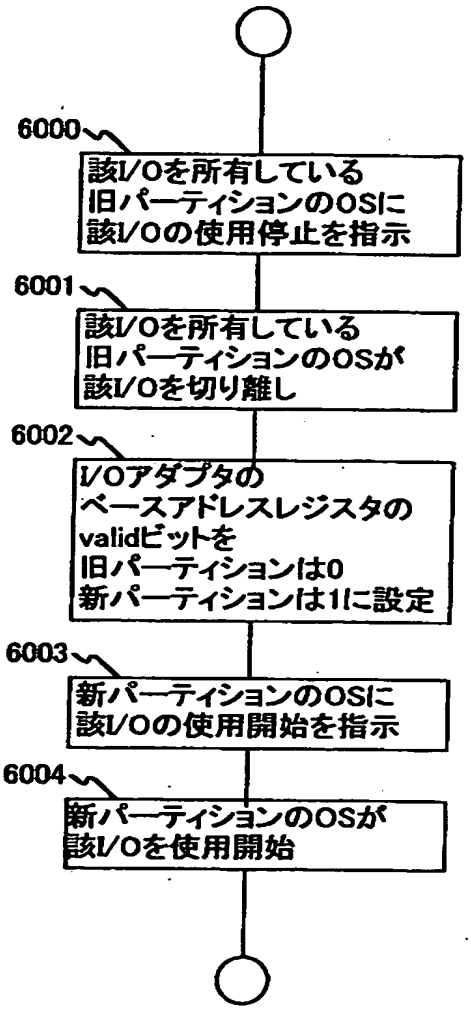
【図１５】

図15


パーティション0
送信性能モニタリング
カウンタ
2100
パーティション1
送信性能モニタリング
カウンタ
2101
パーティション2
送信性能モニタリング
カウンタ
2102
パーティション3
送信性能モニタリング
カウンタ
2103
パーティション0
受信性能モニタリング
カウンタ
2110
パーティション1
受信性能モニタリング
カウンタ
2111
パーティション2
受信性能モニタリング
カウンタ
2112
パーティション3
受信性能モニタリング
カウンタ
2113

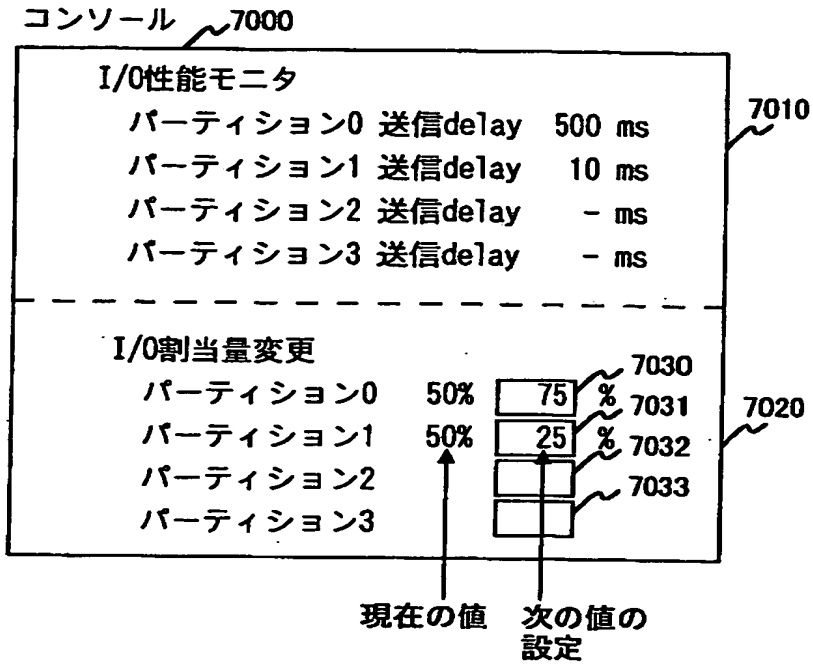
【図１６】

図16


I/Oアダプタのパーティション間移動
6000
該I/Oを所有している旧パーティションのOSに該I/Oの使用停止を指示
6001
該I/Oを所有している旧パーティションのOSが該I/Oを切り離し
6002
I/Oアダプタのベースアドレスレジスタのvalidビットを旧パーティションは0新パーティションは1に設定
6003
新パーティションのOSに該I/Oの使用開始を指示
6004
新パーティションのOSが該I/Oを使用開始

【図１７】

図17



【図１８】

図18

設定ファイル 7050

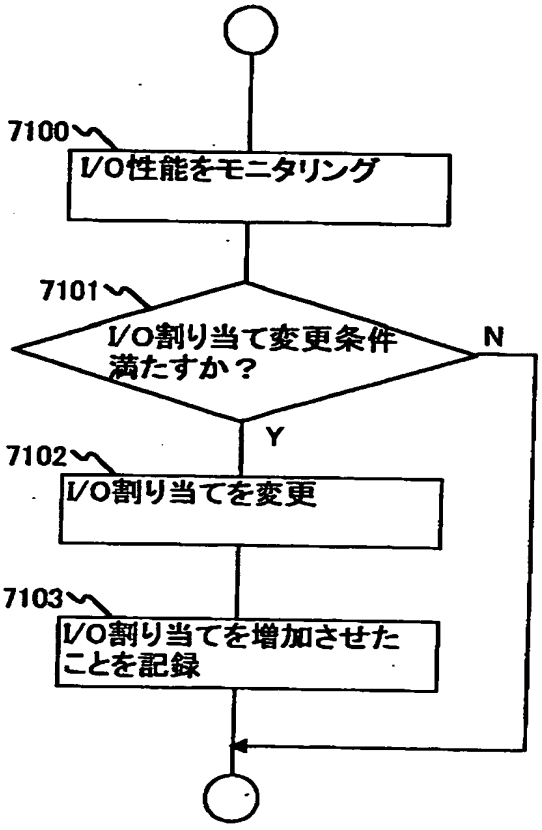
I/O割当量

日時	ﾊﾟｰﾃｨｼｮﾝ0	ﾊﾟｰﾃｨｼｮﾝ1	ﾊﾟｰﾃｨｼｮﾝ2	ﾊﾟｰﾃｨｼｮﾝ3
8:00	50%	50%	0%	0%
18:00	75%	25%	0%	0%

【図１９】

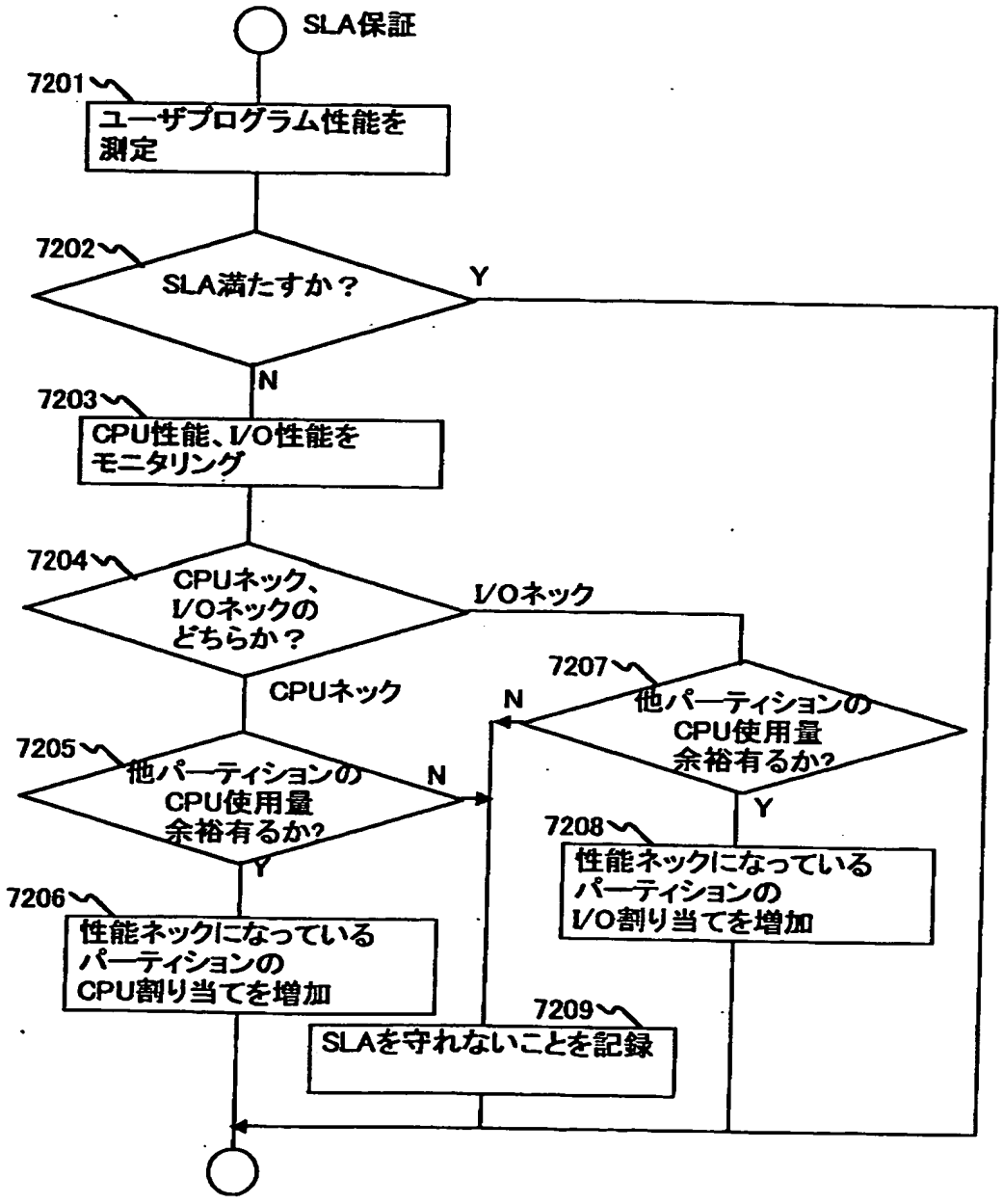
図19

I/O割り当て自動変更



【図２０】

図20


SLA保証
7201
ユーザプログラム性能を測定
7202
SLA満たすか？
Y
N
7203
CPU性能、I/O性能をモニタリング
7204
CPUネック、I/Oネックのどちらか？
I/Oネック
CPUネック
7207
他パーティションのCPU使用量余裕有るか?
N
Y
7205
他パーティションのCPU使用量余裕有るか?
N
Y
7208
性能ネックになっているパーティションのI/O割り当てを増加
7206
性能ネックになっているパーティションのCPU割り当てを増加
7209
SLAを守れないことを記録

【書類名】要約書

【要約】

【課題】パーティションを持った計算機において、パーティションへのＩ／Ｏ割当てをＣＰＵ割当とは独立に制御する。

【解決手段】各Ｉ／Ｏアダプタおよびパーティションにおいて、Ｉ／Ｏアダプタのパーティションへの割当てを時分割で制御するスケジューリング手段、Ｉ／Ｏアダプタをパーティションに空間分割で割当てる手段、パーティション制御プログラムが上記の割当てを動的に変更する手段を設ける。また、パーティションごとの入出力性能を計測する手段を、パーティションごとの性能に基づいてユーザプログラムのＳＬＡを保持する手段を設ける。

【選択図】図１

認定・付加情報

特許出願の番号	特願２００１－０１５１９６
受付番号	５０１００００９１８５８
書類名	特許願
担当官	第七担当上席　０００９６
作成日	平成１３年　１月２５日

＜認定情報・付加情報＞

【提出日】	平成13年 1月24日

次頁無

出 願 人 履 歴 情 報

識別番号　〔０００００５１０８〕

１．変更年月日　１９９０年　８月３１日
［変更理由］　新規登録
住　所　東京都千代田区神田駿河台４丁目６番地
氏　名　株式会社日立製作所